Runway Operations Planning:
A Two-Stage Solution Methodology

Ioannis Anagnostakis, yianag@mit.edu

Prof. John-Paul Clarke, johnpaul@mit.edu

Dept. of Aeronautics & Astronautics, Massachusetts Institute of Technology

77 Massachusetts Avenue, Room 35-217, Cambridge, MA 02139, USA

Tel.: 617-253-3507

ABSTRACT

The airport runway is a scarce resource that must be shared by different runway operations (arrivals, departures and runway crossings). Given the possible sequences of runway events, careful Runway Operations Planning (ROP) is required if runway utilization is to be maximized. Thus, Runway Operations Planning (ROP) is a critical component of airport operations planning in general and surface operations planning in particular. From the perspective of departures, ROP solutions are aircraft departure schedules developed by optimally allocating runway time for departures given the time required for arrivals and crossings. In addition to the obvious objective of maximizing throughput, other objectives, such as guaranteeing fairness and minimizing environmental impact, may be incorporated into the ROP solution subject to constraints introduced by Air Traffic Control (ATC) procedures. Generating optimal runway operations plans was approached in [2] with a "one-stage" optimization routine that considered all the desired objectives and constraints, and the characteristics of each aircraft (weight class, destination, Air Traffic Control (ATC) constraints) at the same time. Since, however, at any given point in time, there is less uncertainty in the predicted demand for departure resources in terms of weight class than in terms of specific aircraft, the ROP problem can be parsed into two stages. In the context of the Departure Planner (DP) research project, this paper introduces Runway Operations Planning (ROP) as part of the wider Surface Operations Optimization (SOO) and describes a proposed "two stage" heuristic algorithm for solving the Runway Operations Planning (ROP) problem. Focus is specifically given on including runway crossings in the planning process of runway operations. In the first stage, sequences of departure class slots and runway crossings slots are generated and ranked based on departure runway throughput under stochastic conditions. In the second stage, the departure class slots are populated with specific flights from the pool of available aircraft, by solving an integer program. Preliminary results from the algorithm implementation on real-world traffic data are included in [1].

1 INTRODUCTION

Unusually high delays have been observed in the departure flow at many major European and US airports. Most of these delays occur at the takeoff queue next to the runway, where aircraft line up with their engines running waiting for a slot on the active runway. Similar delays occur during other phases of the taxi out process, i.e. before the aircraft reaches the takeoff queue, when poor planning results in excessively long waits (with the engines running) at intersections and/or ramps. All these delays result in economic (higher fuel costs) and environmental (higher emissions) inefficiencies.

Therefore, in order to mitigate these adverse economic and environmental effects of ground congestion and delays it is critical that:

- Runway efficiency is improved,
- Runway queue delays are minimized,
- Taxi out times are minimized and
- "Engine start" time is controlled (presently left at the pilot's discretion).

Airport departure management includes several control tasks, i.e. pushback, "engine start" time, taxiway entry, runway assignment and takeoff clearances. In many instances, these tasks must be performed under conditions of high workload and time criticality. In addition, observations of operations at airports such as Boston Logan [15],

[16], [17], Washington Dulles [4] and Newark [9], both of which have more hub operations than Logan, indicated that the dynamics of airport ground flows heavily depend on Air Traffic Control (ATC) constraints and how these affect each airport site and contributed to developing an operational framework, as well as proposing a system architecture. Therefore, given this complexity of the departure process and the airport-specific nature of departure operations, it is difficult for controllers to fully explore all the possible solutions within the relatively short time period in which decisions must be made. This raises the need for automated decision-support systems for planning and controlling ground departure flows. Such systems will automatically explore a very large number of possible future departure schedules and at the same time reduce existing uncertainties by exercising tighter sequencing and scheduling control on each portion of the departure process.

National Aeronautics and Space Administration (NASA)-sponsored research on causes of departure delay [3], [5], [15], [20] all suggest that automation aids that help optimize and control the departure flow would benefit both controllers and aircraft operators. To that purpose, recent research efforts in the field of airport surface operations focus on decision-aiding technology projects, such as the Surface Movement Advisor (SMA) program at Atlanta's airport, which was a joint Federal Aviation Administration (FAA) and NASA project to help airport facilities operate more efficiently [12], [13], [19] and MITRE's DEPARTS project [10], [11]. In fact, the primary objective of the Surface Management System (SMS) research prototype being developed by NASA is to contribute to the understanding and solution of various problems existing on the surface of airports within the National Airspace System [6], [7], [21].

This paper documents research work in the field of departure operations planning being conducted at the Massachusetts Institute of Technology (MIT). The paper is organized as follows: Section 2 describes the structure of the Surface Operations Optimization problem in general and Surface and Runway Operations Planning is specific. In this context lies the introduction and application of the "two-stage" runway operations planning algorithm that is described in Section 3. Sections 3.1 and 3.2 present ideas on possible formulations of the objective functions and constraints for each of the two stages of the algorithm. Section 4 presents an example solution case. In Section 5, a short summary is given, together with topics of future work in this area.

2   PROBLEM STRUCTURE

Departures and arrivals interact through the common use of airport resources (gates, taxiways and runways). Thus, managing the departure flow at an airport requires an integrated "surface-air" solution that considers all the aircraft on the ground as well as the aircraft in the air that are expected to land during the time period when the departures under consideration are still on the airport surface.
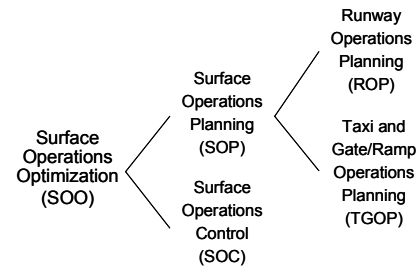

Figure 1: Problem Structure

The tasks involved in what may be described as Surface Operations Optimization (SOO) are depicted in Figure 1. As the figure shows, SOO may be divided into two main tasks:

- **Surface Operations Planning (SOP)** i.e. generating feasible and optimal (or near optimal) plans for distributing available runway time to the different types of operations that require runway time (departures, arrivals and runway crossings).
- **Surface Operations Control (SOC)** i.e. executing the plans in the safest and least workload intensive manner.

As Figure1 also shows, the Surface Operations Planning task may be further sub-divided into two subtasks:

- **Runway Operations Planning (ROP)** i.e. designing the takeoff sequence and schedule while accounting for uncertainties in pushback and taxi operations [10].
- **Taxi and Gate/Ramp Operations Planning (TGOP)** i.e. determining the appropriate taxi and ramp sequence and schedule required to ensure that the takeoff sequence and schedule is materialized.

The algorithmic solution to the ROP problem is a Virtual Queue as proposed in [3] i.e. a virtual extension of the physical takeoff queue that depicts the takeoff sequence for all departures under consideration regardless of their position on the airport surface.

3 TWO-STAGE RUNWAY OPERATIONS PLANNING

The methodology described in [2] was designed to solve the ROP problem in a *"one-stage" optimization routine* that considers all objectives and constraints at the same time. The methodology described in this paper, *parses the runway operations problem into two simpler stages*, as depicted in the flow chart in Figure 2.
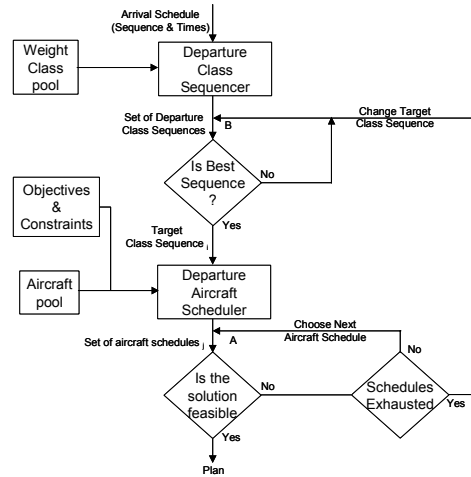

Figure 2: Optimization in stages

The objective of maximizing throughput and all factors that affect throughput, such as wake vortex separation and crossing delay constraints are addressed in the first stage. All other system objectives, such as delay minimization and constraints, such as downstream constraints (splitting departure routes, jet-prop mix, arrival-departure mix), workload limitations and intersecting runways are considered in the second stage.

In the first stage of the solution process, a sequence of time slots that specify the weight class of the aircraft that should occupy a given time window is developed. This "class sequence" is designed to maximize departure throughput. In the second stage of the solution process, aircraft are assigned to the time slots that are developed in the first stage. The resulting "aircraft schedule" is designed to address all the other objectives that were not addressed in the first stage. The objectives and constraints of each stage are explained in more detail below.

3.1 Stage 1

The goal of the first stage is to maximize departure throughput. This is achieved by developing a sequence of departures that minimizes the impact of the constraints that affect the separation between successive departing aircraft.

3.1.1 Constraints

The first constraint that affects the separation between successive departing aircraft is the minimum separation requirement imposed by air traffic control on successive runway operations because of wake vortex considerations. These separation requirements are the set of times and distances that govern the separation between successive departures, successive arrivals, a departure followed by an arrival, and an arrival followed by a departure. For all "leading-trailing" pairs of aircraft, the set of separation requirements is a system parameter that can be input in the planning system as a square matrix, with row coordinates corresponding to all possible weight classes for the leading aircraft and column coordinates corresponding to all weight classes for the trailing aircraft. Each entry of the matrix may be modified by the planner as desired.

Two additional constraints that affect the separation between successive departures and therefore the departure runway throughput are:

- The limit on the number of successive arrivals that may be accommodated on an arrival runway if the arrivals on that runway must cross the active departure runway, i.e. the number of arrivals accommodated between runway crossings must be less than or equal to the capacity for holding arrivals between the arrival and departure runways and
- The maximum delay that an aircraft waiting to cross can absorb.

Both of these constraints affect departure runway throughput by affecting the times when departures will have to be interrupted in order for crossings to cross the active departure runway. The limit values for both of these constraints (max number of aircraft or max number of delay minutes) can be input to the planning system as system parameters the value of which can easily be adjusted in real time by the planner.

3.1.2 Objective Function

The objective of maximizing throughput can be translated to minimizing the time when the latest operation is cleared to use the runway. The formulation of the objective function is as follows: Let N_A be the total number of arrivals and N_D the total number of departures considered. Then, $N_A + N_D = N$, is the total number of "mixed" operations on the runway(s) during the current scheduling window. Therefore, maximizing departure throughput can be achieved by minimizing the time of the last takeoff:

Max departure throughput:
$$\min \max t_{D_i}, \; where \; 1 \le i \le N_D,$$

and maximizing "total" throughput can be achieved by minimizing the time of the last "runway operation" (departure, arrival or crossing):

Max aggregate throughput:
$$\min \max t_i, \ where \ 1 \le i \le N_A + N_D$$

3.1.3   Departure Class Sequencer

The core of the first stage is the Departure Class Sequencer. One of the basic assumptions in this module is that the arrival schedule (sequence and touchdown times) is known in advance. Depending on the runway geometry and inter-dependence, some or all the expected arrivals, after landing and deceleration, become runway-crossing requests on another active runway. The times of these requests can be estimated based on the weight classes of the arriving aircraft and the taxiway space constraints at the specific airport. In many instances, the runway that these arrivals must cross is a departure runway. Thus, maximizing throughput requires appropriate sequencing of departures and runway crossings.

3.1.4   Output

The output of the first stage is a matrix CS of class sequences that are listed in order of throughput i.e. each row is a class sequence and the row number reflects the ranking of the specific sequence relative to the other sequences. Thus, the best sequence is listed in the first row. The matrix is therefore of the form:

CS = [Class Sequence 1,

. . . .

Class Sequence i,

. . . .

Class Sequence m]

3.2 Stage2

The first (best in throughput) of the class schedules from the matrix output CS of the first stage becomes the Target Class Schedule (TCS). The goal of the second stage is then to assign specific aircraft to the weight class slots in the TCS while meeting all or as many of the other constraints that are placed on the departure process. If the selected TCS cannot yield feasible solutions, the next best member of CS is set as the Target Class Schedule. The second stage optimization is formulated as an integer program that assigns specific aircraft to each class slot. The decision variables selected for the formulation are X_{ij}, where $X_{ij} = 1$ if aircraft i occupies slot j, and $X_{ij} = 0$ otherwise.

3.2.1   Constraints

One of the most fundamental constraints in the assignment of aircraft to slots is the requirement that

no aircraft is assigned to a slot that it cannot physically fill i.e. the slot is earlier in time than the earliest time the aircraft can reach the runway. For example, if the earliest time that aircraft i is expected to be at the runway is time 900 and the time at the midpoint of the first two slots in the TCS is earlier than time 900, aircraft i cannot be allowed to occupy slots 1 and 2 in the final solution. This type of constraint can be easily formulated as $X_{ij} = 0$, for j = 1,2.

The class *slot sequence* of the Target Class Schedule also has to be satisfied. Therefore, if for example, aircraft 1 is a large, it can only occupy large class slots in the TCS. This can be guaranteed by setting the constraint $\sum_j X_{1j} = 1, \ \forall \ slot \ j \in L$ where L is the set of large class slots in the Target Class Schedule.

Furthermore, each aircraft must occupy only one slot $\sum_{j=1}^{N_S} X_{ij} = 1, \ \forall \ aircraft \ i$, where N_S is the total number of slots in the class slot sequence, and each slot must be occupied by only one aircraft $\sum_{i=1}^{N_D} X_{ij} = 1, \ \forall \ slot \ j$.

Operational constraints, such as an Expected Departure Clearance Time (EDCT) or a Departure Sequencing Program (DSP), restrict the time that an aircraft can be released for takeoff:

$$tEDCT_{i_1} \le t_{Di} \le tEDCT_{i_2} \quad \text{or}$$
$$tDSP_{i_1} \le t_{Di} \le tDSP_{i_2},$$

where $tEDCT_{i_1}$, $tEDCT_{i_2}$, $tDSP_{i_1}$ and $tDSP_{i_2}$ are the time values that determine the EDCT time window (typically a 15-minute window [17]) or the DSP time window (typically a 3-minute window [17]) as defined by ATC for flight i. Assuming that the expert input of air traffic controllers is available, a heuristic methodology can be inferred to translate a takeoff time window to a takeoff slot window. The takeoff position of each aircraft can be written as a function of the decision variables $\sum_{j=1}^{N_S} j * X_{ij}$ and the above constraints can then be formulated in the form of an acceptable slot range, as follows:

$$sEDCT_{i_1} \le \sum_{j=1}^{N_S} j * X_{ij} \le sEDCT_{i_2} \quad \text{or}$$

$$sDSP_{i_1} \le \sum_{j=1}^{N_S} j * X_{ij} \le sDSP_{i_2}$$

where $sEDCT_{i_1}$, $sEDCT_{i_2}$, $sDSP_{i_1}$ and $sDSP_{i_2}$ are the

takeoff slot end values, as defined by ATC for flight *i*, that define the EDCT or DSP takeoff slot window.

Lifeguard flights or other type of *priority constraints* can be similarly modeled in the form of an upper bound X_{maxTO_i} on the takeoff sequence position:

$$\sum_{j=1}^{N_S} j * X_{ij} \leq X_{maxTO_i}$$

or in terms of inequality constraints between different flights:

$$\sum_{j=1}^{N_S} j * X_{ij} \leq \sum_{j=1}^{N_S} j * X_{kj} .$$

At many airports, localized *sequencing constraints* also affect the departure efficiency. For example, back-to-back departures to the same departure fix are generally not allowed because they require additional gaps between flights. Typically these gaps are achieved by alternating jet and propeller aircraft departures on the same runway, because these two different types of aircraft usually use different departure fixes after takeoff. Such constraints can also be introduced in the form of a position constraint (acceptable departure slot positions for each flight).

Among the most frequently used ATC operational constraints are Miles In Trail (MIT) and (less frequently) Minutes In Trail (MinT) constraints that impose aircraft separations en route. They can be stated in terms of time separation at the takeoff point:

$$\left| t_{D_i} - t_{D_j} \right| \geq \Delta T_{ij}$$

where ΔT_{ij} is the minimum time separation at the takeoff point between flights *i* and *j*, which have an In-Trail restriction, imposed on them. This means that aircraft *i* and *j* can only take off at least ΔT_{ij} time units apart in order to ensure that the In-Trail separation is not violated when they will be airborne. More conveniently for this model, MIT or MinT constraints can be stated in terms of a minimum required takeoff sequence position separation ΔX_{ik} between flights *i* and *k*, which have an In-Trail restriction, imposed on them:

$$\left| \sum_{j=1}^{N_S} j * X_{ij} - \sum_{j=1}^{N_S} j * X_{kj} \right| \geq \Delta X_{ik} \Leftrightarrow \begin{cases} \sum_{j=1}^{N_S} j * (X_{ij} - X_{kj}) \geq \Delta X_{ik} \\ \sum_{j=1}^{N_S} j * (-X_{ij} + X_{kj}) \geq \Delta X_{ik} \end{cases}$$

This means that aircraft *i* and *k* must take off at least

ΔX_{ik} takeoff slots apart from each other to ensure that the In-Trail separation is not violated when they become airborne.

In many cases, maintaining departure fairness among airport users is a difficult task for air traffic controllers. One possible way to achieve fairness is to introduce a "fairness" constraint through the use of a *"Maximum takeoff Position Shifting"* (MPS) constraint that limits the deviation from a "First Come (Call Ready for Pushback) First Serve (Release to Take Off)" policy, unless specific agreements (known to the optimization planning tool) exist between ATC and the airlines. The MPS value may be predetermined by ATC and the airlines. Based on scheduled or "expected to call ready" pushback data, an expected pushback sequence is formed and each aircraft will therefore have its own pushback sequence number. The MPS value then determines the range of acceptable takeoff sequence positions for each departure. For every aircraft i, if X_{PBi} is its pushback sequence position and X_{TOi} is its takeoff sequence position, the MPS value is used in the following constraint:

$$\left| X_{PB_i} - X_{TO_i} \right| \leq MPS \Leftrightarrow \begin{cases} -\sum_{j=1}^{N_S} j * X_{ij} \leq MPS - X_{PB_i} \\ \sum_{j=1}^{N_S} j * X_{ij} \leq MPS + X_{PB_i} \end{cases}$$

where MPS and X_{PBi} are constants that are known in advance.

3.2.2 Objective Function

The main objective in the second stage is to minimize departure aircraft delay i.e. minimize the time that aircraft spend on average taxiing to the runway, subject to all of the above constraints that apply in each particular planning situation.

Given that throughput maximization is addressed in the first stage of the algorithm, a delay-based objective function is used to address the remaining constraints. The time assigned to each runway event is set equal to the midpoint of the time slot to which the specified aircraft is assigned.

For the general case of a runway that serves all types of operations and alterations to the arrival schedule are permitted, let the original arrival (touchdown) times be TOn_i, the projected crossing request times of those arrivals be TX_i and the target departure (clearance to takeoff) times be the class slot midpoint values $TOff_j$ that are calculated. For every arrival i, $1 \leq i \leq N_A$, where N_A is the total number of arrivals considered and for every departure j, $1 \leq j \leq N_D$, where N_D is the total number of departures

considered. $N_A + N_D = N$, is the total number of "mixed" operations on the runway(s) during the current scheduling window. If only departures and crossings are serviced on the runway, then $N_A = 0$.

The delay for each operation is defined as the difference between actual touchdown, crossing or takeoff time and the corresponding earliest possible values for each flight EOn_i, EX_i and $EOff_j$. The latter are calculated using the input arrival and departure schedules and estimated unimpeded taxi time values. Hence, the delay value for each operation represents how much later than its earliest possible time an operation will occur. The total delay for the runway (i.e. minimum arrival, departure and crossing delay) can then be formulated as:

Min aggregate delay:

$$\min\left(\sum_{i=1}^{N_D}\left|TOff_i(x_i) - EOff_i\right|^{k_D} + \sum_{j=1}^{N_A}\left|TOn_j - EOn_j\right|^{k_A} + \sum_{m=1}^{N_A}\left|TX_m - EX_m\right|^{k_X}\right)$$

$$where \;\; 1 \le i \le N_D \; and \; 1 \le j, m \le N_A$$

Minimize ONLY departure delays:

$$\min \sum_{i=1}^{N_D}\left|TOff_i(x_i) - EOff_i\right|^{k_D}, where \;\; 1 \le i \le N_D \;,$$

x_i is the slot position of aircraft i and k_A, k_D and k_X are parameters used to penalize delays of specific flights, with $k_A \ge 1, k_D \ge 1 \; and \; k_X \ge 1$.

3.2.3 Departure Aircraft Scheduler

The core of the second stage is the Departure Aircraft Scheduler. This module develops aircraft schedules for the "Target Class Sequence" (TCS) or best (in terms of throughput) class sequence from the first stage. For each weight class in the TCS, feasible permutations are tested among all the available departing aircraft of that same weight class, in an effort to generate departure aircraft schedules which satisfy all or as many as possible of the remaining system objectives (e.g. delay, environmental impact, fairness). Some of the aircraft schedules that are generated may be unacceptable if they violate system "hard" (inviolable) constraints[1], such as ATC restrictions. If the first (optimal) schedule is not feasible, then the next available aircraft schedule is chosen (feedback A in Figure 2) and stage (b) is repeated. If all the aircraft schedules are exhausted i.e. none of them is feasible, the Target Class Sequence is changed (feedback B in Figure 2) by replacing it with the next available class sequence from the CS matrix.

---

[1] For a definition of "hard" constraints, see [1]

3.2.4 Output

The output of the second stage is a matrix AS of aircraft schedules that are listed in order of their objective value i.e. each row is an aircraft schedule and the row number reflects the ranking of the specific schedule relative to the other schedules. Thus, the best schedule is listed in the first row. The matrix is therefore of the form:

AS = [Aircraft Schedule 1,
………..
Aircraft Schedule j
………..
Aircraft Schedule n]

3.3 Properties of the Two-Stage Solution

At the most fundamental level, both stages perform the two functions required to determine the optimal sequence. The class of each departure slot is defined in the first stage, and specific aircraft are assigned to each of the defined class slots in the second stage. While the second stage may be performed immediately after the first, the two stages may also be performed separately depending on the needs of the particular real-world situation. For example, assume that both stages of the algorithm have been performed and a schedule with specific aircraft for each class slot has been generated. If one or more of these aircraft have difficulty meeting that schedule, the class slot sequence generated by the first process can be left untouched if it is too costly or impractical to change it, while the second stage (aircraft assignment) can be performed independently to assign new flights to substitute for those aircraft that are unable to meet their class slots. Thus, the time scale and level of control in each stage is well matched to the dynamics of the Runway Operations Problem.

In addition, because the throughput is determined in the first stage and the aircraft are assigned to time slots in the second stage assigned, time-based system constraints, such as Estimated Departure Clearance Time (EDCT) slots introduced by ATC, or last minute schedule adjustments to accommodate passenger connections, can be directly incorporated in the optimization with having to consider the impact of these time constraints on throughput.

4 SOLUTION FOR EXAMPLE AIRPORT

To evaluate the potential benefits of improved runway operations planning, the two-stage ROP algorithm was partially implemented for the airport shown in Figure 3, a runway geometry that is frequently encountered at airports.
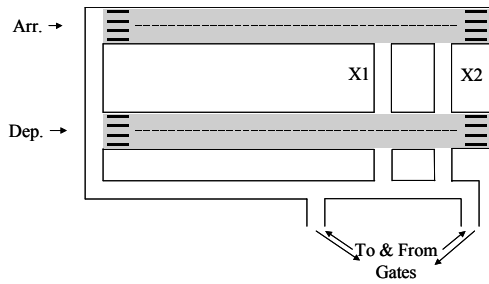
Figure 3: Example airport system

The figure depicts a hypothetical airport system with two parallel runways, one dedicated to arrivals and the other to departures. Using as an input, the departure aircraft classes in hand and the cross-point capacities, the first stage of the solution procedure is to design the best departure sequence i.e. the departure sequence that maximizes runway utilization (throughput).

4.1 Stage 1

Assume that the first six available departures in the "pool" of available aircraft are three small aircraft (S), two large (L) and one heavy (H) and the order they called ready for pushback is: S - S - H - S - L - L. Under a First Come (Call for Pushback) First Serve (Clear to Take Off) control policy, there will be no modifications in the takeoff sequence:

S (200) - S (260) - H (320) - S (440) - L (500) - L (560) - N (560 + 60 = **620**)

where N is the next aircraft taking off right after the last aircraft of this departure group, which will be separated from the last large (L) aircraft by 60 sec whether it is a heavy, a large or a small departure and the numbers in parentheses are the scheduled takeoff times, given the required separations (in seconds) between each pair of aircraft based on their weight classes. It takes a total of 420 seconds for this sequence of six (6) aircraft to complete their takeoffs. However, if the heavy aircraft is positioned last in the takeoff queue, it may lead to throughput savings (or in the worst case no throughput loss) because it requires the largest takeoff separation among all aircraft weight classes (120 sec if followed by a small or large and 90 sec if followed by another heavy aircraft). In that case, the six aircraft can complete their takeoffs in at most 420 sec, or in 390 sec (if the next aircraft N is a heavy) saving 30 sec of runway time:

S (200) - S (260) - S (320) - L (380) - L (440) - H (500) - N (500 + 90 or 120 = **590 or 620**)

Note that, placing a heavy at the end of the sequence may unnecessarily penalize heavies in the

final solution. As the following example will show, it often makes better sense to place a heavy just before a runway crossing. If the arrival stream is taken into consideration, arriving aircraft introduce requests to cross the departure runway and therefore a certain amount of time on the departure runway has to be used for these aircraft to cross. It is possible then, that the heavy departing aircraft need not be the last one to take off in order to achieve maximum throughput on the departure runway. For example, assume that the arrival sequence for five (5) landings cannot be altered and is (as supplied by TRACON):

S (210, 0.5) - S (270, 0.5) - L (330, 1) - S (390, 0.5)

where the numbers in parentheses are the scheduled touchdown times (separations are assumed to be 60 sec between each pair of aircraft) and the taxiway capacity that each aircraft is expected to occupy depending on its weight class (0.5 for small, 1 for large and 1.5 for heavy aircraft). In this example, we assume that only crossing point X2 can be used with a total taxiway capacity of two (2) units. If no arrival schedule modifications are allowed, before all taxiway capacity is used, X2 will accommodate arrivals in the sequence:

S (210, 0.5) - S (270, 0.5) - L (330, 1)

At this point, all three aircraft must be cleared to cross in order to allow for the following small aircraft S (390, 0.5) to land and occupy taxiway space at X2 after clearing the arrival runway.

Assuming that the last aircraft to land (before crossings are cleared) will occupy the runway for 50 sec, all three crossings will be available to cross at time point 330 + 50 = 380. Based on interviews with air traffic controllers, it is assumed that the time necessary for crossings to be completed is 40 sec for the first and 10 sec for each aircraft following. Therefore, the three crossing aircraft will cross the departure runway in 40 + 10 + 10 = 60 sec.

Consequently, if the departure sequence is left unchanged (S - S - S - L - L - H), at time point 380, the first large aircraft of this sequence will take off occupying the departure runway for an (assumed) period of 50 sec and then the same runway will need to be blocked for an additional 60 sec for the crossings. Therefore, the next large aircraft of the departure sequence will be allowed to take off 50 + 60 = 110 sec later at time point 380 + 110 = 490 and the final departure sequence will be (Case 1):

S (200) - S (260) - S (320) - L (380) - **X** - L (490) - H (550) - N (550 + 90 or 120 = **640 or 670**)

where X represents the time point when crossings are cleared and N is the next aircraft taking off right after the last aircraft of this departure group, which will be separated from the last heavy (H) aircraft by 90 sec if

it is also a heavy or by 120 sec if it is a large or a small departure.

Given that the wake vortex separation behind the large aircraft is 60 sec, there is an additional 110 – 60 = 50 sec that is added to the departure schedule due to the crossings and the heavy departure that still remains to be served (take off) incurring at least 90 sec of wake vortex separation to the departure runway. Therefore, in total, there is at least an additional 50 + 90 = 140 sec added to the departure schedule of the seven aircraft (the six original and aircraft N) in Case 1, if crossings are considered.

On the other hand, assume that the original departure schedule of the first six aircraft is changed from S - S - **H** - **S** - L – L to S - S - **S** - **H** - L – L and at time point 380 the heavy aircraft is cleared to take off ahead of the two large aircraft. Under the same assumptions of 50 sec runway occupancy times and 60 sec for three crossings to be completed, the next time a departure will be allowed to take off is again at time point 490. However, 120 sec of wake vortex separation have to be allowed between the heavy and the following large. The advantage in this case is that the heavy aircraft has already been serviced at this point and the departure sequence is (Case 2):

S (200) - S (260) - S (320) - H (380) - **X** - L (500) - L (560) - N (**620**)

The following aircraft N is then able to take off only 60 sec after the last large in order to maintain the required wake vortex separation. Therefore, in Case 2, after crossings are considered, there is still at least 640 – 620 = 20 sec of runway-time savings compared to Case 1 (higher departure throughput). This happens, because the heavy aircraft of this departure group was not left last to take off and the 120 sec of wake vortex separation time behind it was used for crossings.

In the above examples, the problem was decoupled and departures were isolated from arrivals. It was assumed that the arrival schedule cannot be altered and arrivals were simply treated as additional (but fixed) requests for runway time on runways that are used for mix operations. However, changes in the arrival schedule are possible. In fact, the example below will demonstrate that weight class sequence planning can offer benefits to crossing aircraft that may have to absorb less delay at crossing points, if modifications to the landing sequence and timing are allowed.

Arriving flights reach the runway in a certain sequence that is predetermined by TRACON controllers. At some airports this sequence is determined by advanced Air Traffic Management technologies (e.g. CTAS). If no changes are permitted, arriving aircraft inevitably limit the amount of runway time that is left to be allocated to departures and crossings. On the other hand, allowing changes in the arrival schedule may provide flexibility in producing ROP solutions (runway operations schedules) with higher departure and arrival throughput and thereby enable solutions that are closer to optimality. A flexible arrival schedule may also be particularly useful in the event that the heuristic algorithm described above cannot reach a feasible solution. In such a case, adjusting the arrival schedule may help the algorithm to produce a feasible runway operations schedule.

It might be possible to design the optimization heuristics so that the algorithm alters the arrival schedule before solution infeasibility is reached. This can happen by incorporating information about the arrival stream into the optimization algorithms and by taking into account the types of aircraft expected to arrive and request crossing time from the departure runway, the airport geometry and the taxiway capacity constraints. Consequently, crossing operations can become "smarter" and the runway schedule results can be closer to runway throughput optimality. The following example demonstrates one of the possible ways in which the arrival schedule can be linked to crossing and departure operations on a different runway.

In the example airport system of Figure 3, there is limited taxiway space for holding aircraft on two connecting taxiway segments between the two runways (X1 and X2 in Figure 3). The maximum number of aircraft allowed between the runways is predetermined (this can be a simulation test parameter) for each crossing point and depends on the weight class of the aircraft present. Initially, we assume that all small (S) aircraft can exit the runway early enough to make it to cross-point X1 and that all other aircraft (large and heavies) use the other point X2. In some instances, such an assumption may be relaxed in the interest of "smarter crossings."

Using as an input, the arrival aircraft classes in hand and the crossing point capacities, the problem is to design an arrival sequence that brings arrivals to the crossing points in such a way that no cross-point capacity is wasted due to saturation of another crossing point. For example, assume that:

- Small aircraft occupy one half (0.5) unit capacity, large occupy one (1) and heavies occupy one and a half (1.5) units,
- Both cross-points have a capacity of two (2) units and
- The arrival sequence for landings (as supplied by TRACON) is:

S (210, 0.5) - S (270, 0.5) - L (330, 1) - S (390, 0.5) - H (450, 1.5)

where the numbers in parenthesis are the scheduled touchdown times (one minute apart from each other) and the taxiway capacity that each aircraft is expected to occupy. If no arrival schedule modifications are allowed (Case 1), under the assumption made earlier, cross-point X1 will accommodate all small aircraft in the sequence:

S (210, 0.5) - S (270, 0.5) - S (390, 0.5)

with a utilized taxiway capacity of one and a half (1.5) units out of a total of two (2). Cross-point X2 will then have to accommodate one large and one heavy aircraft in the sequence:

L (330, 1) - H (450, 1.5)

However, when the large aircraft lands, there will be no taxiway space left for the heavy (1+1.5 > 2). Positioning the large ahead of the heavy saturates point X2 earlier and therefore one half (0.5) capacity unit in X1 (1.5 < 2) and one (1) capacity unit in X2 (1 < 2) are wasted for this group of arrivals.

Linking the Departure Planner decision-aiding tool and the resulting departure schedules to the arrival stream can improve runway and taxiway space utilization. If, for example (Case 2), the arrival sequence is altered from:

S (210, 0.5) - S (270, 0.5) - **L (330, 1)** - S (390, 0.5) - **H (450, 1.5)**

to:

S (210, 0.5) - S (270, 0.5) - **H (330, 1.5)** - S (390, 0.5) - **L (450, 1)**

and we also assume that small aircraft can and will roll to cross-point X2, then the two cross-points can receive aircraft in the following order:

Cross-point 1: S (210, 0.5) - S (270, 0.5) - L (450, 1)

Cross-point 2: H (330, 1.5) - S (390, 0.5)

without wasting taxiway capacity at all. The "swapping" between the large and the heavy aircraft of the group allows for all arriving aircraft to be accommodated in the taxiway space between the two parallel runways and in that way all five aircraft of them can be crossed at the same time, with the total crossing time being exactly the same as in Case 1 and under the same crossing clearance. However, in this case, the final departure throughput is higher, because the stream on the departure runway is interrupted for crossings only once, as opposed to Case 1, which needed the departure stream to be interrupted twice for all five arrivals to complete their crossings.

The examples presented above illustrated the departure runway throughput benefits that may be achieved when crossing aircraft are included in the planning process. In addition, the two planning examples that include arrivals (with or without changes to the arrival schedule) illustrate the advantage of solving the broader planning problem that includes all kinds of operations on the same runway.

In terms of actually calculating throughput values for each class schedule, the stochastic nature of ground operations leaves no choice but to calculate stochastic throughput using probabilistic distributions for the pushback and taxi processes. Using as a "base" schedule one of the departure class schedules with crossings, these distributions help determine the probability of a class slot actually being at the position it has in the "base" schedule, as opposed to occupying one position up or down in the sequence (shifts of only one position was used for simplicity). For each "base" schedule, its final stochastic throughput is calculated as the expected throughput over all the possible schedules that can be derived from the "base" by performing feasible class slot shifts up or down. Here is an example:

Assume that there are nine (9) departures within the predetermined planning window. Also, assume that it has been estimated that these departing aircraft are expected to interact with four (4) arrivals, which will request runway time in order to cross the departure runway and therefore, one of the departure class schedules including crossings (lowercase letters) is:

S – S - H - **s/s/l** – S - L – L - L - s –L – S

In this schedule, which is considered to be the "base" schedule, there are only four (4) possible class slot (one-position) swaps that can actually affect the throughput of this sequence (Figure 4 - X1 and X2 are abbreviations for the two crossings groups).
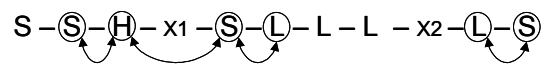

Figure 4: Possible class slot swaps that affect throughput

S – S – H – x1 – S – L – L – L – x2 – L – S
S – S – H – x1 – S – L – L – L – x2 – **S – L**
S – S – H – x1 – **L – S** – L – L – x2 – L – S
S – S – H – x1 – **L – S** – L – L – x2 – **S – L**
S – **H – S** – x1 – S – L – L – L – x2 – L – S
S – **H – S** – x1 – S – L – L – L – x2 – **S – L**
S – **H – S** – x1 – **L – S** – L – L – x2 – L – S
S – **H – S** – x1 – **L – S** – L – L – x2 – **S – L**

Figure 5: Schedules derived from the "base" schedule by performing all possible swap combinations

Taking all possible combinations of occurrence of these four swaps, the set of possible class schedules that can be derived from the "base" schedule consists of sixteen (2^4) schedules (including the "base"). A sample part of this set is shown in Figure 5.

The throughput for each of those derived schedules is calculated based on pushback and taxi time probabilistic distributions. For each schedule, the mean value for the start time of the first class slot is the mean "Time at the Runway" for the earliest aircraft in the departure pool, that has the same weight class as the starting class slot (in this example, the earliest large (L) aircraft). Assuming that the pushback process and the remainder of the taxi process are independent from each other, this mean runway time is calculated as the sum of the mean pushback time (including pushback delays) and the mean taxi time (from gate to runway threshold) for that specific aircraft and for the specific terminal it is coming from. The stochastic distributions were initially assumed to be normal. Their parameters (mean and standard deviation) can either take values derived empirically from real-world data that were collected at Boston's Logan airport, or values derived from curve fitting to Airline Service Quality Performance (ASQP) data, as in [18].

Once the start time is determined, the start times for the rest of the class slots are easily determined based on wake vortex separation criteria. After that, each class slot can have a probabilistic curve associated with it. The overlapping regions between curves of adjacent class slots, determine the probability of a swap between those two slots occurring. Based on those swap probabilities and on the combination of swaps involved in each derived schedule, a probability of occurrence and a throughput value for that particular derived schedule can be calculated. The final stochastic throughput for the "base" schedule is calculated as the expected throughput over the throughput values of all the derived schedules, each of them considered with its individual probability of occurrence.

This process is repeated for each class schedule with crossings, and finally the list is ordered according to throughput in descending order. The first few departure class schedules with crossings are then considered to be the best in terms of maximizing throughput and therefore, they are candidates to become the "Target Class Schedule" in the second stage of the algorithm.

4.2 Stage 2

As stated previously, the goal of the 2nd stage is to assign specific aircraft to the weight class slots defined in the 1st stage. Assuming that the Target Class Schedule in Figure 4 is the Target Class Schedule from the 1st stage, then the goal is to assign each of the nice (9) aircraft under consideration to a class slot. If the time that a given aircraft will take off is assumed to be equal to the midpoint of the slot to which it is assigned and the time that the same aircraft could have take off if there were no departure queues is equal to the time it is ready for pushback plus its unimpeded taxi time, then the delay for each aircraft would be equal to the difference between the midpoint of the assigned slot and the time that aircraft could have take off. This therefore will form the basis for calculating values of the objective function. Thus, if the earliest time that any large aircraft from the departure group can be at the runway is 670, then, based on wake vortex separations between class slots and on landing and crossing runway occupancies, the following set of times corresponds to the start times for the class slots in the Target Class Schedule:

$$670 – 730 – 790 – (X) - 910 – 970 – 1030 – 1090 – (X) - 1190 – 1250$$

And, the following set of times corresponds to the midpoints of the time slots in the Target Class Schedule, which are used for objective function calculations:

$$700 – 760 – 820 – (X) - 940 – 1000 – 1060 – 1120 – (X) - 1220 – 1280$$

There are many feasible ways to assign the nine scheduled departing flights to the nine class slots of the Target Class Schedule and in fact some of them will lead to the same final runway throughput (time to complete all departures) and the same total departure delay.

Table 1 gives two of those feasible, departure-delay-minimizing "aircraft to class slot" assignments that the second stage optimization generates based on two slightly different sets of constraints. In both cases, the same ATC operational constraints are satisfied in addition to all other physical and "slot sequence" constraints that are present in the problem. However, in one case, there is no MPS constraint (fairness is not enforced), while in the other case, there is a constraint of at most 3 position shifts between pushback and takeoff (MPS = 3). The difference (position shift) between columns 4 and 6 for the "MPS = 3" case, evidently shows that in the second case the MPS constraint is satisfied. However, the difference between column 5 (No MPS) and column 6 (MPS = 3) of Table 1 shows that in order for the MPS constraint to be satisfied, the optimal takeoff slot assignment has to be changed.

Even though this is the case, fairness is achieved while the final throughput and total departure delay remain the same.

Flight Number	Time	Weight Class	Push Back Pos.	Take off Pos. (No MPS)	Take off Pos. (MPS = 3)	Max Delay (min)
USC168	670	S	1	1	1	1
COA339	730	S	2	9	5	1
AAL1317	790	H	3	4	4	2
Crossings	*850*	*s/s/l*				
N109FX	910	S	4	3	3	3
DAL1821	970	L	5	2	2	5
SGR501	1030	L	6	8	9	3
USA1854	1090	L	7	6	7	4
Crossings	*1140*	*s*				
USS6171	1190	L	8	7	8	5
N180M	1250	S	9	5	6	6

Table 1: Example "aircraft to class slot" assignments

5   SUMMARY - FUTURE WORK

This paper introduced a "two stage" optimization algorithm for solving the Runway Operations Planning (ROP) problem i.e. to determine the optimal departure schedule. The sole objective of the first stage is to determine the best (from a throughput perspective) departure class sequence (including runway time for crossing operations) to be used in the second stage. The second stage of the optimization algorithm is formulated as an integer program that generates a solution that represents the assignment of aircraft to class slots. Given that throughput maximization is addressed in the first stage of the algorithm, a delay-based objective function is used to address the remaining constraints. Fairness and ATC considerations are introduced into the formulation as constraints.

The "two-stage" algorithm was implemented using Matlab and Simulink. While the Matlab model is not yet complete, it has sufficient functionality to evaluate the fundamental behavior of both stages in the optimization algorithm. Preliminary results from this model can be found in [1].

Apart from the obvious future work of modeling other airport geometries and exploring issues associated with executing the runway operations plans that are developed, there are several model parameters worth exploring. These include the:

- Length of the planning window and the resulting number of aircraft (departures and arrivals) included in the planning window.
- Crossing point (taxiway) capacity and maximum

crossing delay constraints. They affect the location and length of the crossing "gaps" that must be introduced into the departure schedule.

- Probability distributions for the pushback and taxi processes. They affect the stochastic throughput calculations of class schedules and the fidelity of the model.

REFERENCES

1. Anagnostakis, I., Clarke, J-P., "Runway Operations Planning: A Two-Stage Heuristic Algorithm", presented at the *AIAA Aircraft, technology, Integration and Operations Forum*, Los Angeles, CA, October 1st – 3rd, 2002

2. Anagnostakis, I., Clarke, J-P., Böhme, D., Völckers, U., "Runway Operations Planning and Control: Sequencing and Scheduling," *AIAA Journal of Aircraft*, November-December 2001, Vol. 38, No.6, also presented at the 34th *Hawaii International Conference on System Sciences*, Maui, Hawaii, January 3rd – 6th, 2001

3. Anagnostakis, I., Idris, H. R., Clarke, J-P., Feron, E., Hansman, R.J., Odoni, A.R., Hall, W.D., "A Conceptual Design of a Departure Planner Decision-Aid," International Center for Air Transportation, MIT, *AIAA Progress in Aeronautics and Astronautics: Air Transportation Systems Engineering*, American Institute of Aeronautics and Astronautics, Vol. 193, Chapter 21, Washington, District of Columbia, also at the *3rd USA/Europe International Air Traffic Management R & D seminar*, ATM-2000, proceedings, session E: Airport, Ground Operations, Arrival and Departure Management, Naples, Italy, 13-16 June 2000, sponsored by FAA & Eurocontrol, available at http://atm-seminar-2000.eurocontrol.fr/ [cited Oct. 6, 2000]

4. Anagnostakis, I., Carr, F., Evans, A., Idris, H., "Preliminary Observations at Washington Dulles Airport (IAD), May 2001", Massachusetts Institute of Technology, 2001, available at http://icat-server.mit.edu/~fcarr/Publications/IAD_Prelim.ppt. [cited May. 22, 2002]

5. Andersson, K., Carr, F., Feron, E., Hall, W., "Analysis and Modeling of Ground Operations at Hub Airports", *presented at the 3rd USA/Europe Air Traffic Management R&D Seminar*, Naples, Italy, 13rd - 16th June 2000

6. Atkins, S. and Hall, W., "A Case for Integrating the CTAS Traffic Management Advisor and the Surface Management System", *AIAA Guidance,*

Navigation, and Control Conference, Denver, CO, Aug. 14-17, 2000

7. Atkins, S. and Brinton, C., "Concept Description and Development Plan for the Surface Management System", *Journal of Air Traffic Control, Vol. 44, No. 1, January – March, 2002*

8. Carr, F., Evans, A., Clarke, J-P., Feron, E., "QMAST: Queuing Model for Airport Surface Traffic", Massachusetts Institute of Technology, 2002, available at http://icat-server.mit.edu/~fcarr/online_model/ [cited April 2002]

9. Carr, F., Evans, A., Clarke, J-P., Feron, E., "Modeling and Control of Airport Queuing Dynamics under Severe Flow Restrictions." *American Control Conference 2002 proceedings (ISBN 0-7803-7299-9)*, Anchorage, Alaska, May 8-10, 2002

10. Cooper, W.W., Mohleji, S.C., Burke, C.D., Foster, G., Mills, M.J., "Determination of Minimum Push-Back and Taxi-Out Time Predictability Needed For Near-Term Departure Scheduling using DEPARTS" *presented at the 4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, New Mexico, USA, 3rd - 7th December 2001

11. DEPARTS research project summary at: http://www.mitre.org/technology/mtp01/quads_charts/decision_support/cooper.shtml

12. Glass B.J., Gawdiak, Y., "Integrated Navigation Applications in Airport Surface Traffic Management," *4th International Conference on Integrated Navigation Systems*, St. Petersburg, Russia, May 1997

13. Glass B.J., "Automated Data Exchange and Fusion for Airport Surface Traffic Management," AIAA 97-3679, *proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, Aug. 11-13, 1997, pp. 904 – 911

14. Idris, H.R., Delcaire, B., Anagnostakis, I., Hall, W.D., Hansman, R.J., Feron, E., Clarke, J-P., Odoni, A., "Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools," 2nd USA/Europe International Air Traffic Management R & D seminar, ATM-98, Orlando, FL, 1 – 4 December 1998

15. Idris, H.R., Anagnostakis, I., Delcaire, B., Hansman, R.J., Clarke, J-P., Feron, E., Odoni, A.R., "Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools," *Air Traffic Control Quarterly Journal*, Special Issue in Air Traffic

Management, Volume 7, Number 4, 1999, pp. 229-257

16. Idris, H.R., Delcaire, B., Anagnostakis, I., Hall, W.D., Pujet, N., Feron, E., Hansman, R.J., Clarke, J-P., and Odoni, A.R., "Identification of Flow Constraint and Control Points in Departure Operations at Airport Systems," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, Aug. 1998, Vol.2, pp.947-956

17. Idris, H.R., "Observation and Analysis of Departure Operations at Boston Logan International Airport," *PhD. Thesis*, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 2000.

18. Idris, H.R., Clarke, J-P., Bhuva, R.R., Kang, L.S., Queuing Model for Taxi-Out Time Estimation, *Air Traffic Control Quarterly Journal*, Volume 10, Number 1, 2002

19. Lawson, D. R., "Surface Movement Advisor (SMA)", Journal of Air Traffic Control, January-March, 1998

20. Pujet, N., Feron, E., "Input-Output Modeling and Control of the Departure Process of Congested Airports", Air Traffic Control Quarterly Journal, Volume 8, Number 1, pp. 1 – 32, 2000

21. Surface Management Systems (SMS) research summaries can be found at:
a) http://ffc.arc.nasa.gov/our_projects/sms.html
b) http://www.metronaviation.com/airport-surface-mgt.html